Blockchain Mining Games

Blockchain Mining Games*

Aggelos Kiayias† Elias Koutsoupias ‡ Maria Kyropoulou‡
Yiannis Tselekounis†

November 20, 2016

Abstract

We study the strategic considerations of miners participating in the bitcoin's protocol. We formulate and study the stochastic game that underlies these strategic considerations. The miners collectively build a tree of blocks, and they are paid when they create (mine) a node which will end up in the path of the tree that is adopted by all. Since the miners can hide newly mined nodes, they play a game with incomplete information. Here we consider two simplified forms of this game in which the miners have complete information. In the simplest game the miners release every mined block immediately, but are strategic on which blocks to mine. In the second more complicated game, when a block is mined it is announced immediately, but it may not be released so that other miners cannot continue mining from it. A miner not only decides which blocks to mine, but also when to release blocks to other miners. In both games, we show that when the computational power of each miner is relatively small, their best response matches the expected behavior of the bitcoin designer. However, when the computational power of a miner is large, it will deviate from the expected behavior, and other Nash equilibria arise.

1 Introduction

Bitcoin is the most successful decentralized digital currency. It was first presented in the white paper [Nakamoto, 2008] under the pseudonym Satoshi Nakamoto. Its backbone is the blockchain protocol which attempts to keep a consisted list of transactions in a peer-to-peer network. The blockchain protocol successfully solves the real distributed problem of agreement, and has the potential to support novel applications which require distributed computing across a network.

Game-theoretic issues are very important for the correct execution of the blockchain protocol. This was realized at its inception when its creator, Nakamoto, analyzed incentives in a simple, albeit insufficient, model. Understanding these issues is essential for the survival of bitcoin and the development of the blockchain protocol. In practice it can help understand their strengths and vulnerabilities and, in economic and algorithmic theory, it can provide an excellent example for studying how rational (selfish) players can play games in a distributed way and map out their possibilities and difficulties.

Distilling the essential game-theoretic properties of blockchain maintenance is far from trivial; some "attacks" and vulnerabilities have been proposed but there seems to exist no systematic way to discover them. In this work, we study two models in which the miners (the nodes of the network that run the protocol and are paid for it) play a *complete-information stochastic game*. Although the miners in the actual blockchain game do not have complete information, our games aim to capture two important questions that selfish miners ask: (a) what to compute next

*This work is supported by ERC Advanced Grant # 321171 (ALGAME), Horizon 2020 grant # 653497, (PANORAMIX), ERC Starting Grand # 249184 (CODAMODA).

†University of Edinburgh. Email:{Aggelos.Kiayias,Ioannis.Tselekounis}@ed.ac.uk

‡University of Oxford. Email:{elias,kyropoul}@cs.ox.ac.uk

(more precisely, which block to mine) and (b) when to release the results of computation (more precisely, when to release a mined block). By considering only complete information games, we may weaken the immediate applicability of the results, but we obtain a clean framework for studying these issues with proper focus and in a rigorous way.

We consider two stochastic games: (a) the immediate-release game in which every miner reveals immediately the blocks that he mines; in this game, the strategy of every miner is to select an appropriate block to mine and, (b) the strategic-release game in which miners not only can select which block to mine, but they can also withhold releasing blocks; we add the interesting twist that they must immediately announce a successful mining of a block but not the block itself; in this way, all miners have complete information of the current situation but can mine only the blocks which have been completely released by their discoverers. This interesting twist turns a very complicated game of incomplete information into an attractive, highly non-trivial game with complete information. We believe that although this is not the game actually modeling the real world[1], it is of great value in understanding the game-theoretic aspects of the highly convoluted incomplete information game that takes place in the real world.

1.1 The Bitcoin currency and the blockchain protocol

The Bitcoin currency came into existence in 2009 when the first ₿50 (bitcoins) were introduced to the system. Since then it has gained significant recognition with more than ₿15 million currently in circulation with an exchange rate of ₿1 for around 500€ at the time of this writing.

Bitcoin was the first successful design of a fully distributed currency and is based on a peer-to-peer network achieving consensus on the broadcasted bitcoin transactions. The difficulty in the design of a fully distributed digital currency lies in avoiding the possibility of double spending the (easily reproducible) digital coins, [Nakamoto, 2008, Finney, 2011, Karame et al., 2012]. Eliminating these possibilities is far from trivial in a decentralized setting, with applications that go way beyond the monetary nature of bitcoin. A practical pure consensus protocol can have far-reaching implications in many domains, e.g., secure distributed timestamping [Haber and Stornetta, 1991] or decentralizing Internet name services, (see also [Pease et al., 1980] [Okun and Barak, 2007] regarding the topic of Byzantine Agreement in the standard and anonymous version respectively).

Bitcoin's design is based on several previous attempts. The idea is that the users maintain a public log of all transactions that have taken place between the clients (bitcoin owners). The history of the recorded transactions alone determines the ownership of the bitcoins, so it is imperative that the users reach an agreement about it. Perhaps the most important element of the Bicoin protocol, allowing it to provide a consensus solution in a setting where identities are not available, is the notion of *proof of work* [Dwork and Naor, 1993] [Back, 1997] [Juels and Brainard, 1999] [Rivest et al., 1996]. Bitcoin uses cryptography to make the miners provide proof of work before validating a block of transactions in an operation that is reminiscent of Sybil attack prevention techniques that were discussed earlier, see e.g., [Aspnes et al., 2005]. Cryptographic techniques (digital signatures) are also used to guarantee that only the rightful owner of bitcoins can "spend" them by including them in a transaction.

We now give a brief but detailed description of how the Bitcoin protocol works. As mentioned before, the goal is for the users to maintain a public ledger listing all transactions of the form $\{X \textit{ pays } Y \textit{ the amount of } ₿Z\}$ in a distributed fashion, i.e., without the need for a central authority. This is called the *blockchain*. To add a block of transactions to the blockchain and claim the corresponding reward, a user has to solve a hard cryptographic puzzle, thus spending computational power. This process is called mining and we will use the term *miner* to refer to the users.

[1]It is interesting to point out that there are potential real world settings where the fact that a block was mined becomes known but the block itself may not released. This would correspond to a setting that a head of a pool withholds a block, but a pool participant that found the block announces its discovery.

Ideally, the blockchain would be a simple chain of blocks implying precedence between the corresponding transactions, i.e., a serialization of valid transactions between the clients of the Bitcoin protocol. This would be the case if miners always started mining at the last announced block and propagated each block creation immediately to the network of the remaining miners. However, the selfish nature of the miners who try to receive the rewards of as many blocks as possible (or even the inherent delay of block propagation in the distributed network)[2] can result in temporary forks in the blockchain. The protocol suggests to the miners to always start mining at the end of the branch which needed the largest amount of computational effort so far, i.e., the end of the longest fork. This strategy is called FRONTIER and we will call the miners that follow it *honest* miners.

In order to mine a block, the miners are asked to compute some nonce value such that adding that value to the hash of the preceding block (specifying the branch of the blockchain that they are extending) and the batch of transactions they are trying to validate will render a block whose hash (SHA-256) does not exceed a certain threshold. Solving this puzzle is assumed to be computationally hard; however, the verification of a provided solution is easy. It is also assumed that in order to solve the puzzle, a miner cannot do any better than trying it for different inputs repeatedly. If a miner has more computational power than another, then he has higher probability to solve the puzzle faster and create the block. The difficulty of the puzzles is adjusted so that a single block is created every 10 minutes on average. Once a miner solves a puzzle and creates a block, he becomes eligible to receive a reward. It is important for our game-theoretic analysis that the reward is given only if the corresponding block is permanently added to the blockchain. The reward for each successful mining is a fixed amount of newly created bitcoins plus fees from the transactions that are included in the block. This fixed reward was originally set to ₿50 while the protocol determines that it will be halved each time $210,000$ blocks are permanently added to the blockchain, which happens approximately every 4 years. This implies that the protocol will eventually stop creating new bitcoins (10^{-8} BTC is the minimal unit of Bitcoin), at which time the fixed rewards for mining will be entirely replaced by transaction fees.[3]

The reward structure of the protocol guarantees that the honest miners' revenue is proportional to their computational power. However, understanding when it is profitable for the miners to deviate from the honest strategy is a central question and has attracted a lot of attention. The original assumption was that no miner has an incentive to deviate from the honest strategy if the majority of the miners are honest. However, this is not true as was shown by Eyal and Sirer [Eyal and Sirer, 2014]. They gave a specific strategy which, when followed by a miner with computational power at least 33% of the total power, provides rewards strictly better than the honest strategy (assuming that every other miner is honest). This was extended computationally in [Sapirstein et al., 2016].

1.2 Our results

We consider two stochastic games whose states are rooted trees. The nodes of the tree are blocks that have been mined in the past. At every time-step, each miner selects a node of the current tree and tries to extend it by one new block. The probability that a miner succeeds in mining a new block is proportional to the miner's computational power. The utility of a miner is the fraction of successfully mined blocks in the common history.

In the *immediate-release* game, the miner can select any node of the tree to mine, while in the *strategic-release*, the miner can select only nodes that have been (declared) *released* by their creator.

If a miner has computational power above a threshold, he may not mine a node at the

[2] Accidental forking happens every 60 blocks on average.

[3] Currently, the transaction fees are insignificant compared to fixed rewards (approximately 1.33%).

frontier (the set of deepest nodes) in the hope that his blocks will become the accepted history instead of the already mined nodes. Also, he may not release a node immediately in the hope that the computational power of the other miners will be wasted in mining nodes which will not be part of the common history.

Our work seeks to identify thresholds of the computational power below of which FRONTIER (the honest strategy), is a Nash equilibrium while above them it is no longer the optimal strategy. We denote by h_0 and $\hat{h}_0$ the threshold for the immediate-release and the strategic-release games, respectively. It is easy to see that for both games $h_0, \hat{h}_0 \in [0, 0.5]$. Our results are as follows.

- We prove that in the immediate-release setting the threshold is $0.361 \leq h_0 \leq 0.455$. This implies that a miner with at most 36% of the total computational power can not gain more than 36% of the total rewards, i.e., his fair share which he would gain by being honest and following the suggested strategy FRONTIER; and that a miner with computational power more than 46% will always deviate from the honest strategy. An immediate consequence is that if every miner has computational power less than h_0, then FRONTIER is a *Nash equilibrium*. We have experimentally determined that the actual threshold is close to $h_0 \approx 0.42$.

- Regarding the strategic-release setting, we prove rigorously that the corresponding threshold is lower bounded by $\hat{h}_0 \geq 0.308$ (root of the polynomial $p^3 - 6p^2 + 5p - 1$). A similar result was obtained recently by Sapirshtein, Sompolinsky, and Zohar in [Sapirstein et al., 2016]. Their result is tighter and sets the threshold below 0.33 (cf. Figure 2 in their paper for $\gamma = 0$). The difference between our approach and the approach of [Sapirstein et al., 2016], is that they provide an algorithm to compute an approximately optimal strategy and then run their algorithm to estimate the threshold. The authors acknowledge the fact that their "algorithm copes with computational limitations by using finite MDPs as bounds to the original problem, and by analyzing the potential error that is due to inexact solutions". Our result on the other hand, gives an exact, albeit suboptimal bound on the threshold and considers exact best responses in a purely mathematical (not computational) way.

Open problems: This work sets the stage and makes progress towards a systematic study of these complicated stochastic games. A lot of issues remain open. For example, besides the obvious problem of tightening our results, there are a lot of interesting questions about the Nash equilibria above the thresholds h_0 and $\hat{h}_0$ in both types of games.

1.3 Related work

The Bitcoin protocol was originally introduced in [Nakamoto, 2008] and was built based on ideas from [Back, 1997] and [Dai, 1998]. After the "creation" of Bitcoin several other alternative electronic currencies followed, known as *altcoins* (http://altcoins.com/), e.g., litecoin, Primecoin etc. The Bitcoin white paper provides a probabilistic analysis of double spending attacks while a more detailed analysis can be found in [Rosenfeld, 2014]. Bitcoin's design and research challenges are discussed in [Bonneau et al., 2015] along with a presentation of the existing research. In [Tschorsch and Scheuermann, 2015] an extensive and more introductory survey on distributed cryptocurrencies can be found.

The works most relevant to ours are [Kroll et al., 2013], [Eyal and Sirer, 2014] and [Sapirstein et al., 2016]. In [Kroll et al., 2013], the equilibria of the Bitcoin game are considered. The authors observe that any *monotonic* strategy is a Nash equilibrium (one of many). Their analysis though is quite restricted: as we show, even in the case of the immediate release game, FRONTIER is not best response for values above 0.455. In [Eyal and Sirer, 2014], the authors prove that a guaranteed majority of honest miners is not enough to guarantee the security of the Bitcoin protocol. In particular they present a specific strategy called the "Selfish Mine" strategy and

examine when this strategy is beneficial for a pool of miners. It appears that a fraction of $1/3$ of the total processing power is always enough for a pool of miners to benefit by applying the Selfish Mine Strategy no matter the block propagation characteristics of the network ($\gamma = 0$ in their setting). Hence, this constitutes a profitable attack against the Bitcoin protocol. Furthermore, when block propagation is favoring the attacker the threshold above which "Selfish Mine" is beneficial is any non-zero value. It is not hard to see that the "Selfish Mine" strategy does not fully exploit settings where block propagation is more favorable to the attacker. This was observed in [Garay et al., 2015] where an optimal attack against the property of chain quality was considered in their setting where the network is considered to be completely adversarial. In [Sapirstein et al., 2016] the authors consider a wider set of possible strategies that includes the "Selfish-Mine" strategy and explore this space computationally. Their analysis also accounts for possible communication delays in the network, the presence of which can diminish the profit threshold.

A vast majority of previous work examines possible types of attacks against the Bitcoin protocol and suggest adaptations of the protocol to ensure its security. We very briefly mention some of these works here.

Successful pool mining related attacks are discussed in [Rosenfeld, 2011] and [Courtois and Bahack, 2014]. In [Eyal, 2014] the author considers attacks performed between different pools where users are sent to infiltrate a competitive pool giving raise to a *pool game*. See also [Lewenberg et al., 2015] for a (cooperative) game theoretic analysis regarding pool mining. [Babaioff et al., 2012] deals with information propagation and Sybil attacks. The authors propose a reward scheme which will make it in the best interest of a miner to propagate the transactions he is made aware of and not duplicate. [Kroll et al., 2013] considers an attack that can be performed from people that are only interested in destroying Bitcoin, as opposed to other attacks performed by users trying to increase their expected reward. This is called the *Goldfinger* attack. [Heilman et al., 2015] focuses on the peer-to-peer network and examines *eclipse* attacks where the attacker(s) isolates a node/user from the network and forces him to waste his computational power thus participating in an attack without even being aware. Certain deanonymization attacks have also recently been observed [Meiklejohn et al., 2013] by analysing the transactional graph (see also [Fergal Reid, 2012] and [Ron and Shamir, 2013]).

In [Sompolinsky and Zohar, 2015] an alternative consensus method is described, called Greedy Heaviest-Observed Sub-Tree or GHOST. A variant of GHOST has been adopted by Ethereum, a distributed applications platform that is built on top of block chains. [Eyal et al., 2014] attempts to overcome scalability issues that arise in Bitcoin (block size and interval vs latency and stability) by proposing a new scalable blockchain protocol. [**?**] provides another alternative for scalability that utilizes off chain transactions while using the distributed ledger for maintaining the contracts between the parties that engage in off chain transactions. In [Garay et al., 2015] the authors analyze the Bitcoin protocol in depth. They abstract the core of the protocol that they term the *bitcoin backbone* and prove formally its main attributes and properties, which can be used as building blocks for achieving goals other than simply maintaining a public ledger. They show that when the propagation delay in the network is relatively small, an honest majority of users is enough to guarantee smooth operation in a cryptographic model where the rationality of the players is not considered.

2 The Bitcoin Mining Game and its Variants

The game-theoretic issues of bitcoin mining can be captured by the following game-theoretic abstraction. The parameters of the game are:

- the number n of miners or players
- the probabilities $p = (p_1, \ldots, p_n)$ that miners succeed in solving the crypto-puzzle; these

are proportional to their computational power and they sum up to 1: $\sum_{i=1}^{n} p_i = 1$.

- the depth of the game d; the payment for mining a new block is not payed immediately, but only after a chain of certain number of new blocks is attached to it; in the current implementation of the Bitcoin protocol this number is $d = 100$. We will mainly consider games with $d = \infty$, but we will discuss briefly how they are affected by this parameter.

Two more parameters could play a role in a more general model of the protocol. The computational cost c^* of mining a new block and the reward (payment) r^* per block accepted by all miners (currently approximately 25 bitcoins). Here we assume that the reward r^* is constant and we scale all payments so that $r^* = 1$. Also, if the expected gain is high enough to entice a miner to participate[4], its actual value is not important, since the miner tries to maximize revenue.

AK: Note that because of the distributed nature of the Bitcoin protocol, it is possible that more than one miners succeed almost simultaneously to mine a new block. We choose to ignore this aspect here; nevertheless, it is easy to generalize our model to a setting where $\sum_{i=1}^{n} p_i < 1$ and thus there is a non-zero probability at each step that no miner will be awarded a block.

During the execution of the protocol, the miners build a tree of blocks to which they try to add more blocks. The protocol aims to increase the height of this tree by one every ten minutes, on average. Once a miner succeeds in creating a block, the new node is added to the tree. However if the miner is strategic, he may have reasons not to add the newly discovered node to the tree. Therefore besides the publicly known tree, each miner might have his own private tree.

Definition 1 (State)**.** *A public state is simply a rooted tree. Every node is labeled by one of the players. The nodes represent mined blocks and the label indicates the player who mined the block. Every level of the tree has at most one node labeled i because there is no reason for a player to mine twice the same level.*

A private state of player i is similar to the public state except it may contain more nodes called private nodes and labeled by i. The public tree is a subtree of the private tree and has the same root.

In the incomplete information case, the private states may also include the partial knowledge that players have about the other players (knowledge about the probabilities of other private trees, but also about their knowledge etc). This is a very complicated case, and we do not treat it in this work. Instead we treat two complete-information cases in which all miners know the private states of all miners:

Immediate-release model Whenever a miner succeeds in mining a block, it releases it immediately, and all miners can continue from the newly mined block.

Strategic-release model Whenever a miner succeeds in mining a block, it becomes common knowledge. However, the miner may decide to postpone the release of the block. Until the block is released, other miners cannot continue mining from this block, although they are aware of its existence.

AK: While the second model has no counterpart in practice, we believe it is of high theoretical interest as it can serve as an intermediate model between immediate-release and strategic-release with incomplete information. The immediate-release model enables the study of miners that follow the protocol in terms of block propagation but mine strategically, while the strategic release allows us to extend the study to the game-theoretic issues of block witholding. Although ideally we would want to study the latter under the incomplete information regime, we will

[4]The protocol must satisfy Individual Rationality. Rational participating players should have non-negative expected utility.

defer this for future work since the game becomes substantially more complex to analyze. It is important to stress that any strategy in the full information setting is also a valid strategy in the incomplete information setting. Importantly, if a strategy is not dominant in the full information setting it cannot be dominant in the incomplete information setting XXX check this XXX. Remaining of paragraph to be removed. The second model makes little sense in practice, but we believe that it is of high theoretical interest as an intermediate model between immediate-release and strategic-release with incomplete information. The immediate-release model allows us to study the game-theoretic issues of mining while the strategic-release model allows us to extend this study to the game-theoretic issues of withholding the release of blocks. Although ideally we would want to study the latter under the incomplete information regime, it is not clear that we can convincingly model such a game, let alone study and understand it.

We describe the set of strategies for the strategic-release case. The immediate-release case is the special case in which the release function has been fixed to immediate-release.

Definition 2 (Strategy)**.** *A pure strategy of player i consists of two functions (μ_i, ρ_i):*

- *the mining function μ_i which selects a node of the current public state to mine.*
- *the release function ρ_i which is a (perhaps empty) private part of the player's state which is added to the public state.*

  Both functions depend on the state of knowledge of the miner. For the strategic-release case with incomplete information they depend on the public state and the private states of all miners.

The suggested strategy by the designer of the protocol is the FRONTIER strategy.

Definition 3 (FRONTIER)**.** *We say that a miner follows the* FRONTIER *strategy when he releases any mined block immediately and selects to mine one of the deepest nodes.*

In the expected execution of the protocol in which all players play the FRONTIER strategy, the Bitcoin protocol creates a path. AK: This is due to our simplifying assumption that no players simultaneously mine a block. In practice, the FRONTIER strategy creates something very close to a path, with occasional "orphan" blocks hanging from it.

The game is played in phases. In each phase, each player i uses his mining function μ_i to select a block to mine. We assume that exactly one player succeeds in mining a block in each phase, and that the probability of success for each player is given by the probabilities $(p_1, \ldots, p_n)$. The winner then adds the newly mined block to his private tree as a leaf hanging from his mined node. He then applies his release function which may add some of his private part of the tree (for example, the newly mined node) to the public tree. This may trigger a cascade of releases from other players. When the dust settles, we will have a new public tree, and each player will have updated knowledge[5]. The phase ends at this point and a new phase begins.

Note that it is possible that the release function of the winning player may result in an empty release. Since here we consider the complete information case, all miners can immediately detect the end of the phase. In the incomplete information case however this is not possible, although the miners can estimate the probability of this happening, and that adds another complication in modeling strategic considerations in the incomplete information regime.

Payments for mining new blocks are essential to incentivize the players to try to mine new blocks. A miner who succeeds in mining a block is paid d phases later (currently $d = 100$); the delay is considered sufficient to guarantee that no long branches off the main path exist. The description of the payment scheme seems sufficient *under the assumption that branches become*

[5]The releasing step is non-deterministic and, depending on the release functions and the order of applying the release functions, may lead to different outcomes. However, this never happens in the cases we analyze here.

stale quickly and that only the main trunk survives. With the term *trunk* we refer to a long path with ignored stale branches (i.e., the sibling of a paid node as well as its descendants will get no payment at any point in time, thus they are effectively deleted).

A rigorous game-theoretic analysis of the Bitcoin protocol is quite complicated because of the potential strategic branching, and it requires a more precise definition of payments. To be consistent with the non-game-theoretic considerations of the Bitcoin protocol, we assume that at every level (i.e., height of the tree) only one node is paid, the first one which succeeds in having a descendant d generations later. In graph-theoretic terms, a node u is paid when its path from the root is extended by a path of length d; when this happens every sibling (as well as its descendants) of node u becomes stale.

Definition 4 (Payments)**.** *For some nodes of the tree, the miners who discovered them will get a fixed payment (normalized to 1). The payments comply with the following rules:*

- *the nodes that receive payment must form a path from the root. This immediately adds the restriction that at every level of the tree exactly one node receives payment.*
- *among the nodes of a single level that satisfy the above path restriction, the first one which succeeds in having a descendant d generations later receives payment.*

Since only one node per level is paid for, the utility of a miner in the long run is defined as the fraction of the total payment which he receives (his paid nodes over the total number of paid nodes).

When a node is paid, rational miners will completely ignore every branch that starts at an earlier node. So in the long run, the tree essentially becomes the trunk (a long path with ignored stale branches) with a small tree of depth at most d at the end. We will call such a game *truncated at level d*. Immediate-release truncated games are *finite stochastic games*.

We will also consider games that are not truncated at a specific level d. We have to do this with care, since it is possible that two or more miners will continue expanding their own branch forever and they will never agree. However, in our games this cannot happen in an optimal play when one miner has probability less than $1/2$. The reason is that the utility of a miner is the fraction of the total payment he receives which is expected to be 0 if he keeps mining his own branch forever (a case of gambler's ruin).

3 The Immediate-Release Game

In this section we determine the conditions which guarantee that the suggested Frontier strategy is a Nash equilibrium. We fix the strategy of all but one miners to Frontier and identify when Frontier is the best response of the remaining miner.

We can assume that all miners who follow the Frontier strategy by assumption act as a single miner AK: (as mentioned, we consider only the simplified setting where miners do not simultaneously produce blocks). This gives rise to a two-player (two-miner) game: Miner 1 is the miner whose optimal strategy (best response) we wish to determine and has relative computational power p (p fraction of the total computational power), while Miner 2 is assumed to follow the Frontier strategy and have collective relative computational power $1 - p$.

A (public) state is simply a rooted tree of width at most 2. In the immediate-release case, after pruning away stale (abandoned) branches, the state is a long path (called trunk) followed by two branches, one for each miner of lengths a and b (see Figure 1). The lengths of these two branches determine the state and can be 0. Also because Miner 2 plays Frontier, his path must be the longest one, except temporarily when Miner 1 mines a block and moves ahead; in this case we have $a = b + 1$, and when this happens, Miner 2 abandons his path and continues

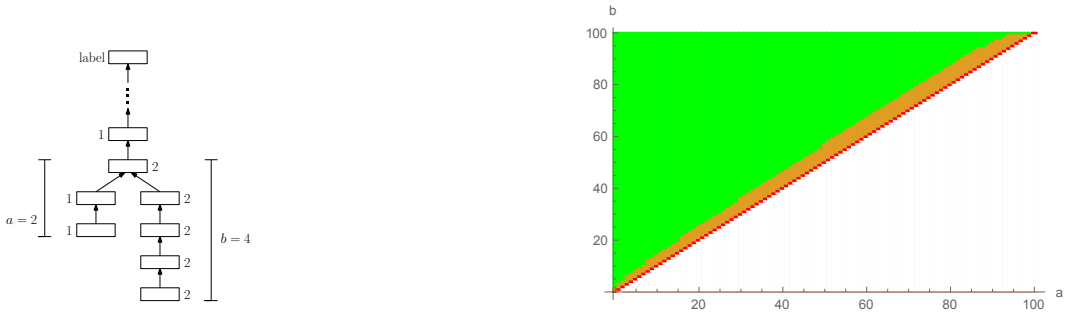

Figure 1: One the left, a typical state (tree). On the right, the set of states form the optimal strategy for the truncated game at $d = 100$. The upper-left green part is the set C of capitulating states, the diagonal red line is the set W of winning states, and the orange part the set M of mining states.

from the frontier of the other path. To summarize, the states of the game are the pairs (a, b) with $0 \leq a \leq b + 1$.

The set of states (a, b) can be partitioned into three parts (see Figure 1):

Mining states: the set M in which both miners keep mining their own branch. State $(0, 0)$ belongs to M. Depending on the strategy of miner 1, additional states might be included in M.

Capitulation states: the set C of states in which Miner 1 gives up on its branch and continues mining from some block of the other branch. When the game is truncated at depth d, this set includes all states of the form (a, d), for $a = 0, \ldots, d$.

Winning states: the set W of states in which Miner 2 capitulates. Given that miner 2 plays FRONTIER it holds that $W = \{(a, a - 1) : a \geq 1\}$.

Note that when Miner 1 capitulates and abandons its own branch, it can choose to move to any state $(0, s)$. Since the miner is rational, it will select the best of these states (we assume that in case of a tie, it will always select the same state). AK: It follows that without loss of generality we can describe fully the strategy of Miner 1 by specifying the set of mining states M and the state $(0, s)$ that is chosen when Miner 1 capitulates. The *set of deterministic strategies* of Miner 1 is exactly the set of pairs $(M,\ s)$, where M is the set of mining states and $(0, s)$ is the landing state to which the miner jumps from any capitulation state.

Let $g_k(a, b)$ denote the expected gain of Miner 1 when the frontier advances by k new levels starting from an initial tree in which the branches of Miner 1 and 2 have lengths a and b, respectively. It should be intuitively clear that, in the long run, the expected gain per level should be almost independent of the initial state. AK: As a result, for large k, k', it should be that $g_k(a, b) - g_{k'}(a, b)$ is independent of a, b and thus it is merely a function of k, k'. Furthermore, assuming that constant rewards are given and expenses per block created remain constant, it will hold that $g_k(a, b)$ satisfies for any large k, k',

$$\frac{g_k(a, b) - g_{k'}(a, b)}{k - k'} = g^*$$

for some constant g^* which represents the *expected gain per level* in the long run. Based on this we can define the expected gain as

$$g_k(a, b) = k \cdot g^* + \varphi(a, b),$$

where the potential function $\varphi(a,b) = \lim_{k\to\infty} g_k(a,b) - k \cdot g^*$ denotes the advantage of Miner 1 for currently being in state (a,b); that this limit exists follows from a straightforward argument[6]. The objective of Miner 1 is to maximize g^*. AK: fix this

For a strategy (M,s), we can define $g_k(a,b)$ recursively as follows. When the current state is (a,b), there are three possibilities:

- If $(a,b) \in M$, both miners continue mining. With probability p, Miner 1 succeeds to mine the next block first and the new state is $(a+1,b)$; with the remaining probability Miner 2 succeeds and the new state is $(a,b+1)$.
- If $(a,b) \in C$, Miner 1 abandons its branch and the new state is $(0,s)$. The trunk increases by $b-s$ blocks.
- If $(a,b) \in W$, Miner 2 abandons its branch and the new state is $(0,0)$. The gain for Miner 1 is $a \cdot g^*$. The trunk increases by a blocks.

The frontier advances when either Miner 2 wins or when Miner 1 wins and $a = b$. From the above consideration, we get

$$g_k(a,b) = \max\left(\max_{s=0,1,\ldots,b-1} g_k(0,s),\ p\begin{cases} g_{k-1}(0,0)+a+1, & \text{if } a=b \\ g_k(a+1,b), & \text{if } a<b \end{cases} + (1-p)\, g_{k-1}(a,b+1)\right) \tag{1}$$

and by definition $g_0(a,b) = 0$. From this, we can get a similar recurrence for φ:

$$\varphi(a,b) = \begin{cases} \varphi(0,0) + a - g^* & a = b+1 \\ \max\left(\max_{s=0,\ldots,b-1} \varphi(0,s), p\varphi(a+1,b) + (1-p)\varphi(a,b+1) - (1-p)g^*\right) & \text{otherwise} \end{cases}$$

We also fix $\varphi(0,0) = 0$; note that the potential of all states is non-negative.

We note that the above definitions do not take depth d into account (consider $d = \infty$). This is without loss of generality for the proof of Theorem 1 where we use this recurrence, as proving that Frontier is best response for $d = \infty$ (i.e., when Miner 1 has a superset of available winning paths than for any constant d) implies that the result also holds for any constant d.

3.1 Frontier is a NE iff $p \le h_0$, where $0.361 \le h_0 \le 0.455$

If every miner plays Frontier, their expected gain is proportional to the probability of mining a block and therefore proportional to their relative computational power p_i. For this we get the following proposition.

Proposition 1. Frontier *is a Nash equilibrium if, having fixed the strategy of all miners except i to* Frontier*, the best response of a miner i has expected gain per level equal to p_i.*

AK: we need proof here. also the iff statement

In this section we bound the threshold on the computational power of each miner so that Frontier is a Nash equilibrium. The main result is the lower bound that follows.

Theorem 1. *In the immediate-release model,* Frontier *is a Nash equilibrium when every miner has relative computational power $p_i \le \frac{1}{3}(1 - 8/(1+3\sqrt{57})^{1/3} + (1+3\sqrt{57})^{1/3}) \approx 0.361$ (root of the polynomial $2p^2 - (1-p)^3$).*

[6]Since state $(0,0)$ is clearly recurrent, we can alternatively define $\varphi(a,b)$ as the expected value $g_k(a,b) - k \cdot g^*$ until state $(0,0)$ is reached.

Starting in a mining state, one of the two miners will eventually capitulate and join the other branch.[7] The probability that the miner wins such a race plays a significant role in our analysis. A formal definition follows.

Definition 5. *Let $r_{M,s}(a,b)$ denote the winning probability starting at state (a,b), that is, the probability that a winning state will be reached before a capitulation state.*

We will use the notation of $r(a,b)$ for the optimal strategy (M, s), and $r_\infty(a,b)$ when the miner never capitulates (i.e., when $C = \emptyset$).

The probability $r_{M,s}(a,b)$ can be defined recursively as follows:

$$r_{M,s}(a,b) = \begin{cases} p \cdot r_{M,s}(a+1,b) + (1-p) \cdot r_{M,s}(a,b+1), & (a,b) \in M, \\ 1, & (a,b) \in W, \\ 0, & (a,b) \in C. \end{cases}$$

The next simple lemma which bounds the probability $r(a,b)$, plays a central role in our analysis:

Lemma 1. *The following holds for every state (a,b):*

$$r(a,b) \leq r_\infty(a,b) = \left(\frac{p}{1-p}\right)^{1+b-a}.$$

Proof. The inequality follows from the fact that the miner has more opportunities/paths to reach a winning state when he never gives up.

Regarding the equality, we essentially want to solve the recurrence for r_∞

$$r_\infty(a,b) = \begin{cases} 1 & \text{when } a = b+1 \\ p \cdot r_\infty(a+1,b) + (1-p) \cdot r_\infty(a,b+1) & \text{otherwise.} \end{cases}$$

Consider the quantity $l = 1 + b - a$ which captures the distance of state (a,b) from the set of winning states. This quantity decreases by one when the miner succeeds in mining a block, and increases by one when the opponent succeeds. Since we consider r_∞, the case in which the miner never gives up, the situation is a gambler's ruin version: the quantity l behaves as the position of a biased random walk on a half-line with an absorbing state at 0, probability p of moving away from the absorbing state, and probability $1-p$ of moving towards the absorbing state. The probability $r_\infty(a,b)$ is the probability of reaching the absorbing state starting at position $l = 1 + b - a$ which can be easily computed to be $(\frac{p}{1-p})^l$. □

The next lemma gives a very simple necessary and sufficient condition for the potential function so that Frontier is a Nash equilibrium. Intuitively, the condition states that Miner 1 capitulates from state $(0,1)$, so no other state except $(0,0)$ will ever be reached.

Lemma 2. *Strategy* Frontier *is a best response for Miner 1 if and only if $\varphi(0,1) = \varphi(0,0)$.*

Proof. From the definition of φ, we have:

$$\begin{aligned} \varphi(0,0) &= p\,\varphi(1,0) + (1-p)\,\varphi(0,1) - (1-p)\,g^* \\ \varphi(0,1) &= \varphi(0,0) + 1 - g^*. \end{aligned}$$

It follows that $\varphi(0,1) - \varphi(0,0) = (g^* - p)/(1-p)$. Since Frontier is best response if and only if $g^* = p$, the lemma follows. AK: we need to invoke a proposition here - be explicit □

[7] This happens even when d is unbounded. For a miner with relative computational power $p_i < 1/2$ who never capitulates, the expected gain is 0, as he is engaged in a gambler's ruin situation. Lemma 1 shows that the probability of winning drops exponentially with the distance of its branch from the frontier.

We now derive a very useful relation between the expected optimal gain of a pair of states and the winning probability of one of them.

Lemma 3. *For every state (a, b) and every nonnegative integers c and k:*

$$g_k(a + c, b + c) - g_k(a, b) \leq c \cdot r(a + c, b + c).$$

Conversely for $p < 1/2$, there is $\epsilon_{a,b}(k)$ which tends to 0 as k tends to infinity, such that

$$g_k(a + c, b + c) - g_k(a, b) \geq c \cdot r(a, b) - \epsilon_{a,b}(k).$$

Proof. We focus on the first inequality since the proof of the second inequality follows from similar reasoning. Suppose that the current state is (a, b) and Miner 1 continues playing not in the optimal way, but by simulating the strategy that he would follow had the current state been $(a + c, b + c)$. The crucial observation is that the simulation can be carried out because the strategy of the other miner, based only on the difference $b - a$, is unaffected.

Let $\hat{g}_k(a, b)$ be the gain for the next k levels using this potentially suboptimal strategy. Let also $\hat{r}$ be the probability that Miner 1 will reach a winning before a capitulating state (within the next k levels). Then we must have

$$\hat{g}_k(a, b) = g_k(a + c, b + c) - c \cdot \hat{r}.$$

Now, since the simulated strategy cannot be better that the optimal strategy, it is clear that $g_k(a, b) \geq \hat{g}_k(a, b)$. Furthermore, the probability $r(a + c, b + c)$, which is the probability that Miner 1 wins the competition even if more than k levels are used, is at least equal to $\hat{r}$. These two bounds give the first part of the lemma:

$$g_k(a, b) \geq \hat{g}_k(a, b) = g_k(a + c, b + c) - c \cdot \hat{r} \geq g_k(a + c, b + c) - c \cdot r(a + c, b + c).$$

The second inequality follows from similar considerations and in particular by starting in state $(a + c, b + c)$ and simulating the strategy as being in state (a, b). The term $\epsilon_{a,b}(k)$ is needed because now we want to bound the probability $\hat{r}$ from below: $\hat{r} = r(a, b) - \epsilon_{a,b}(k)$, where $\epsilon_{a,b}(k)$ is the probability that Miner 1 will win the competition in more than k levels. This probability tends to 0 as k tends to infinity; in particular for $p < 1/2$, this is bounded by the probability that a gambler with initial value a and probability of success $p < 1/2$ will win against a bank of initial value b after $\Theta(k)$ steps. □

The second part of the previous lemma will not be used in this work, but it may prove helpful to tighten our results. The following corollary is a direct consequence of the first part of the lemma.

Corollary 1. *For any state (a, b) and nonnegative integer c*

$$\varphi(a, b) \geq \varphi(a + c, b + c) - c \cdot r(a + c, b + c).$$

The gain $g_k(a, b)$ is clearly increasing in a as having mined more blocks cannot hurt the miner. Therefore we get the following useful fact.

Claim 1. *The potential $\varphi(a, b)$ is non-decreasing in a.*

The following three lemmas use the above results to provide explicit bounds on the potential of states $(1, 2)$, $(0, 2)$, and $(0, 1)$ under certain assumptions on the best response of Miner 1 from these states and his computational power.

Lemma 4. *For every p:*

$$\varphi(1, 2) \leq \frac{2p^2 - p}{(1 - p)^2} + g^* \frac{1}{1 - p}. \tag{2}$$

Proof. Consider the potential of state $(1,1)$. We know that

$$\varphi(1,1) \geq p\varphi(2,1) + (1-p)\varphi(2,1) - g^*(1-p) = p(2-g^*) + (1-p)\varphi(1,2) - g^*(1-p).$$

On the other hand, from Corollary 1 and Lemma 1 we can bound it from above by $\varphi(1,1) \leq \varphi(0,0) + r(1,1) \leq \frac{p}{1-p}$. By putting the two bounds together and eliminating $\varphi(1,1)$ we get the lemma. □

Lemma 5. *For $p < (3-\sqrt{5})/2 \approx 0.382$, if state $(0,2)$ is a mining state, i.e. $(0,2) \in M$, then*

$$\varphi(0,2) \leq \frac{2p^2 - (1-p)^3}{(1-p)^2}. \tag{3}$$

It follows that for $p < \frac{1}{3}(1 - 8/(1+3\sqrt{57})^{1/3} + (1+3\sqrt{57})^{1/3}) \approx 0.361$ (root of the polynomial $2p^2 - (1-p)^3$), state $(0,2)$ is not a mining state.

Proof. To bound $\varphi(0,2)$ we use the bound for $\varphi(1,2)$ from the previous lemma and a bound for $\varphi(0,3)$. To bound $\varphi(0,3)$ we use the monotonicity property of $\varphi(a,b)$ with respect to a and apply Corollary 1 and Lemma 1 to $\varphi(1,3)$:

$$\varphi(0,3) \leq \varphi(1,3) \leq \varphi(0,2) + r(1,3) = \varphi(0,2) + \frac{p^3}{(1-p)^3}.$$

Assuming now that $(0,2)$ is a mining state, we get:

$$\begin{aligned}\varphi(0,2) &= p\varphi(1,2) + (1-p)\varphi(0,3) - g^*(1-p) \\ &\leq p\varphi(1,2) + (1-p)\varphi(0,2) + (1-p)\frac{p^3}{(1-p)^3} - g^*(1-p)\end{aligned}$$

and by solving for $\varphi(0,2)$:

$$\begin{aligned}\varphi(0,2) &\leq \varphi(1,2) + \frac{p^2}{(1-p)^2} - g^*\frac{1-p}{p} \\ &\leq \frac{2p^2-p}{(1-p)^2} + g^*\frac{1}{1-p} + \frac{p^2}{(1-p)^2} - g^*\frac{1-p}{p} \\ &= \frac{3p^2-p}{(1-p)^2} - g^*\frac{(1-p)^2-p}{p(1-p)}.\end{aligned}$$

The coefficient of g^* is negative for $p \leq (3-\sqrt{5})/2$, so we can replace it by p to get the first part of the lemma (recall that g^* is always greater or equal to p).

For the second part of the lemma, it suffices to observe that for $p < \frac{1}{3}(1 - 8/(1+3\sqrt{57})^{1/3} + (1+3\sqrt{57})^{1/3})$ the expression $\frac{2p^2-(1-p)^3}{(1-p)^2}$ is negative. Since the potential cannot be negative, it follows by contradiction that $(0,2) \notin M$. □

Lemma 6. *For $p < (3-\sqrt{5})/2$, if $(0,1)$ is a mining state, then $(0,2)$ is also a mining state.*

$$\varphi(0,1) \leq (1-p)\varphi(0,2) - p\frac{1-3p+p^2}{1-p}. \tag{4}$$

Proof. First we bound $\varphi(0,1)$ if we assume that $(0,1) \in M$:

$$\begin{aligned}\varphi(0,1) &= p\varphi(1,1) + (1-p)\varphi(0,2) - g^*(1-p) \\ &= p(p\varphi(2,1) + (1-p)\varphi(1,2) - g^*(1-p)) + (1-p)\varphi(0,2) - g^*(1-p) \\ &\leq (1-p)\varphi(0,2) - p\frac{1-3p+p^2}{1-p}\end{aligned}$$

where we used the bound for $\varphi(1,2)$ from above and $\varphi(2,1) = 2 - g^*$. Towards a contradiction assume now that $(0,2) \notin M$. That is, the miner capitulates at state $(0,2)$ and moves to either state $(0,1)$ or state $(0,0)$. Since the miner can move to state $(0,0)$ in two steps by first moving to state $(0,1)$, we can assume without loss of generality that it moves to state $(0,1)$ and therefore $\varphi(0,2) = \varphi(0,1)$. By substituting this in the above expression we get

$$\varphi(0,1) \leq -\frac{1-3p+p^2}{1-p} < 0$$

for $p < (3-\sqrt{5})/2$, a contradiction. □

We are now ready to prove the main result of this section. We use the bounds provided by the previous three technical lemmas (together with Lemma 2) to prove that Miner 1 capitulates from state $(0,1)$, and that FRONTIER therefore is his best response.

Proof of Theorem 1. For $p \leq \frac{1}{3}(1 - 8/(1+3\sqrt{57})^{1/3} + (1+3\sqrt{57})^{1/3})$, Lemma 5 establishes that $(0,2)$ is not a mining state. But then from Lemma 6, neither state $(0,1)$ is a mining state, or equivalently, FRONTIER is the best-response strategy of Miner 1. □

3.2 Upper bound

The main theorem of this section, Theorem 1, shows that if all miners have relative computational power $p < 0.361$, FRONTIER is a Nash equilibrium. On the other hand, it is intuitively clear that if a miner has computational power close to $1/2$, it will have some advantage if he does not play FRONTIER, against miners who play FRONTIER. Let h_0 be the maximum relative computational power of miners for which FRONTIER is a Nash equilibrium. Experimental results stated in Table 1 AK: we need some details about how the experimental results were obtained, based on computing the potential φ, show that $h_0 = 0.418$.

Table 1: The threshold for different values of d.

	threshold
$d = 2$	0.5
$d = 3$	0.454
$d = 5$	0.432
$d = 10$	0.422
$d = 15$	0.42
$d = \infty$	0.418

Our work in this section is to estimate the threshold h_0. Providing rigorous lower bounds for h_0—as we do in Theorem 1—does not appear to be easy since it involves a non-trivial Markov decision process. However, it is not too hard to obtain good upper bounds of h_0 as it suffices to come up with a mining strategy that has expected gain g^* greater than p. Here we provide a simple such upper bound which can be directly extended to get an upper bound of h_0 close to 0.418.

Theorem 2. *When Miner 2 plays* FRONTIER*, the best response strategy for Miner 1 is not* FRONTIER *when* $p \geq 0.455$.

Proof. It suffices to consider a fixed mining strategy (M, s) that has expected gain per step g^* greater than p. We consider a mining strategy truncated at $d = 3$, i.e., the miner capitulates at every state (a,b) with $b \geq 3$.

We select $M = \{(0,0),(0,1),(1,1),(2,1),(2,2)\}$ and $s = 1$ and define the potential for this strategy as follows:

$$\begin{aligned}
&\varphi(0,0) = 0, \\
&\varphi(0,1) = \frac{g^* - p}{1-p}, \\
&\varphi(a,b) = \varphi(0,1), && \text{for every } (a,b) \in C \\
&\varphi(a,b) = a - g^*, && \text{when } a = b+1 \\
&\varphi(2,2) = p\varphi(3,2) + (1-p)\varphi(0,1) - g^*(1-p), \\
&\varphi(1,2) = p\varphi(2,2) + (1-p)\varphi(0,1) - g^*(1-p), \\
&\varphi(1,1) = p\varphi(2,1) + (1-p)\varphi(1,2) - g^*(1-p).
\end{aligned}$$

We need to select g^* and verify that this is the correct potential. In particular, we need to verify that for all mining states $(a,b) \in M$ we have $\varphi(a,b) \geq \varphi(0,1)$, which holds when

$$g^* = \frac{p^2(2 + 2p - 5p^2 + 2p^3)}{1 - p^2 + 2p^3 - p^4}.$$

We can also verify that for $p \geq 0.455$, the expected gain g^* is strictly greater than p, which establishes that the strategy (M, s) is a better response than Frontier. □

The strategy employed in the proof of the theorem is optimal for the truncated game at $d = 3$ when $p \approx 0.455$. As we mentioned above, one can compute the optimal strategy for the finite games truncated at $d = 4, 5$ and so on, to get better and better upper bounds. These bounds converge relatively quickly to $h_0 = 0.418$.

4 The Strategic-Release Game

Similarly to the immediate-release case, we wish to identify conditions such that Frontier is a Nash equilibrium. To do so, we again assume that all but one miner, say Miner 1, use the Frontier strategy, and then examine the best response of Miner 1. Since all honest miners select the same block to mine and release it immediately once it is mined, they essentially act as one miner, and it is sufficient to consider the case of two miners.

As in the immediate-release case, the tree created by the two miners has width 2. The trunk is permanently fixed and can be safely ignored; hence, the situation is captured by the two branches of the execution tree in addition to a bit of information regarding each block in Miner 1's branch specifying whether or not Miner 1 has released this block. In particular, this situation can be captured by a triple of numbers a, a_r and b, where a and b denote the number of blocks mined by Miner 1 and the honest miner, respectively (the length of the two branches), while $a_r \leq a$ denotes the number of the released blocks on Miner's 1 branch (note that they are consecutive blocks starting from $(0,0)$). AK: better change the notation a_r e.g., to a_r, r looks too much like a variable Since the honest miner immediately announces the mined blocks, all b blocks on his branch are always released. Also, contrary to the immediate-release case, it can be $a > b + 1$.

Under the assumption that Miner 2 follows the Frontier strategy, we know that if Miner 1 has released $a_r \leq b$ blocks, then Miner 2 will not abandon his path, but if $a_r > b$ then the honest miner will immediately capitulate. So, without loss of generality we can assume that if $a \leq b$ then $a_r = a$ while if $a > b$, then $a_r = b$, otherwise the honest miner would have immediately abandoned his path, and the game would be at state $(a - a_r, 0)$. In other words, we can always consider $a_r = \min\{a, b\}$; hence we can capture the state of the game by the tuple (a, b) as in the immediate-release case.

Let $\hat{g}_k(a,b)$ denote the expected gain of Miner 1 when *the branch of the honest miner in the execution tree is extended by k new levels* starting from an initial tree in which the non-common parts of the paths have length a and b, but Miner 1 has only released the first $\min\{a,b\}$ blocks on his path. It should be intuitively clear that, in the long run, the expected gain per level should be almost independent of the initial state. With this in mind, we can write the expected gain as

$$\hat{g}_k(a,b) = k \cdot \hat{g}^* + \hat{\varphi}(a,b),$$

for some constant $\hat{g}^*$ which is the expected gain per level in the long run. The potential function $\hat{\varphi}(a,b)$ denotes the advantage of Miner 1 for currently being in state (a,b) (and having released $\min\{a,b\}$ blocks).

We can define $\hat{g}_k(a,b)$ as follows:

- If $a \leq b$, Miner 1 has two options: to capitulate or to mine. In the latter case, the next state will be $(a+1,b)$ with probability p, and $(a,b+1)$ with the remaining probability.
- If $a > b$, Miner 1 has one additional option to the previous case: it can release an additional block and lead the game to state $(a-b-1,0)$. When this happens, Miner 2 who plays Frontier capitulates. Note that we allow Miner 1 to repeatedly release blocks, which is equivalent to allowing him to release any number of blocs.

From the above consideration, we get the following optimal gain for Miner 1:

$$\begin{aligned}\hat{g}_k(a,b) = \max\Big\{ &\max_{s=0,\ldots,b-1} \hat{g}_k(0,s), p \cdot \hat{g}_k(a+1,b) + (1-p) \cdot \hat{g}_{k-1}(a,b+1), \\ &\hat{g}_{k-1}(a-b-1,0) + b + 1 \Big\}, \end{aligned} \tag{5}$$

where the last term inside the max applies only when $a \geq b+1$; equivalently we can define $\hat{g}(a,0) = -\infty$ when $a < 0$. The base case of the recurrence is $\hat{g}_0(a,b) = 0$.

As in the case of immediate-release, we define a potential $\hat{\varphi}$ from

$$\hat{g}_k(a,b) = k\hat{g}^* + \hat{\varphi}(a,b),$$

when k tends to infinity. We also note that the definition above does not take d into account (considers $d = \infty$). This is without loss of generality for the proof of Theorem 3 as proving the result for $d = \infty$ is stronger.AK: this needs to be explained / proven

The recurrence for the potential is

$$\begin{aligned}\hat{\varphi}(a,b) = \max\Big\{ &\max_{s=0,\ldots,b-1} \hat{\varphi}(0,s), p \cdot \hat{\varphi}(a+1,b) + (1-p)(\hat{\varphi}(a,b+1) - \hat{g}^* \cdot (1-p), \\ &\hat{\varphi}(a-b-1,0) + b + 1 - \hat{g}^* \Big\}, \end{aligned} \tag{6}$$

where again we define $\hat{\varphi}(a,0) = -\infty$ when $a < 0$. We also fix the value $\hat{\varphi}(0,0) = 0$.

4.1 Frontier is a NE iff $p \leq \hat{h}_0$, where $\hat{h}_0 = 0.308$

In this section we will show the following theorem.

Theorem 3. *In the strategic-release model,* Frontier *is a Nash equilibrium for every miner with $p \leq 0.308$ (root of the polynomial $p^3 - 6p^2 + 5p - 1$).*

Theorem 1 established that for any $p \leq 0.361$ there exists a potential φ for the immediate-release model such that $\varphi(0,0) = \varphi(0,1)$ and $g^* = p$. The main idea of the proof is to extend this potential to states (a,b) such that $a > b+1$. This is possible, for the following reasons:

- for states $(b+1, b)$, when p is small and Miner 1 is one block ahead of Miner 2, there is a high risk if he does not release the block; the risk is that Miner 2 can mine one extra block and then Miner 1 may end up in a stale branch.
- for states (a, b) with $a > b+1$, when Miner 1 is at least two blocks ahead, it is safe not to release any blocks until Miner 2 catches up to within distance one.

With this in mind, we define the following potential

$$\overline{\varphi}(a,b) = \begin{cases} \varphi(a,b), & \text{when } a \leq b, \\ a\lambda - b\mu - c, & \text{otherwise} \end{cases}$$

where $\lambda = \frac{(1-p)^2}{1-2p}$, $\mu = \frac{p^2}{1-2p}$ and $c = \frac{p(1-p)}{1-2p}$. The parameters of the second part are chosen so that $\overline{\varphi}(a,b) = p\overline{\varphi}(a+1,b) + (1-p)\overline{\varphi}(a,b+1) - p(1-p)$. It is important to notice that the two pieces fit together nicely in the sense that for $a = b+1$ we have that $\overline{\varphi}(a,b) = \varphi(a,b) = a\lambda - b\mu - c$, so we could use the following equivalent definition for $\overline{\varphi}$:

$$\overline{\varphi}(a,b) = \begin{cases} \varphi(a,b), & \text{when } a \leq b+1, \\ a\lambda - b\mu - c, & \text{otherwise} \end{cases}$$

To ease the presentation, we will use the following notation, for the potential of states (a, b) with $a \geq b+1$: (a) $\overline{\varphi}_M$ when Miner 1 continues to mine, (b) $\overline{\varphi}_R$ when Miner 1 releases one more block and the other (honest) miner capitulates, and (c) $\overline{\varphi}_C$ when Miner 1 capitulates.

$$\begin{aligned} \overline{\varphi}_M(a,b) &= p\overline{\varphi}_M(a+1,b) + (1-p)\overline{\varphi}_M(a,b+1) - p(1-p) \\ \overline{\varphi}_R(a,b) &= \overline{\varphi}(a-b-1,0) + a - p \\ \overline{\varphi}_C(a,b) &= 0. \end{aligned}$$

Note that when Miner 1 capitulates, he starts mining at one of the states $(0, s)$ where $s \leq b-1$, where by assumption the potential is 0. To prove the theorem, we first show that $\overline{\varphi}$ satisfies the recurrence of the strategic-release potential (6) when $\hat{g}^* = p$. Equivalently, using the above notation

$$\hat{\varphi}(a,b) = \max(\overline{\varphi}_M(a,b), \overline{\varphi}_R(a,b), \overline{\varphi}_C(a,b)). \tag{7}$$

Lemma 7. *The potential $\overline{\varphi}$ satisfies the recurrence* (6) *when $\hat{g}^* = p$ and $p \leq 0.308$ (root of the polynomial $p^3 - 6p^2 + 5p - 1$).*

Proof. We break down the proof into three distinct claims:

Claim 2. *For states (a, b) with $a < b+1$,*

$$\overline{\varphi}(a,b) = \max(\overline{\varphi}_M(a,b), \overline{\varphi}_R(a,b), \overline{\varphi}_C(a,b)).$$

Using the alternative definition of $\overline{\varphi}$, the claim holds trivially, since φ satisfies it, and $\overline{\varphi}_R(a,b) = -\infty$.

Claim 3. *For states (a, b) with $a > b+1$,*

$$\overline{\varphi}(a,b) = \overline{\varphi}_M(a,b) = \max(\overline{\varphi}_M(a,b), \overline{\varphi}_R(a,b), \overline{\varphi}_C(a,b)).$$

This claim also follows directly: $\overline{\varphi}(a,b) = \overline{\varphi}_M(a,b)$ holds by design, and it is easy to verify that $\overline{\varphi}_M$ gives the maximum of the three values:

$$\begin{aligned} \overline{\varphi}_M(a,b) &> a\lambda - b\mu - c - \frac{p(1-p)}{1-2p} = \overline{\varphi}_R(a,b), \\ \overline{\varphi}_R(a,b) &\geq 0 = \overline{\varphi}_C(a,b). \end{aligned}$$

Claim 4. *For states* $(b+1, b)$*:*

$$\overline{\varphi}(b+1,b) = \overline{\varphi}_R(b+1,b) = \max(\overline{\varphi}_M(b+1,b), \overline{\varphi}_R(b+1,b), \overline{\varphi}_C(b+1,b)).$$

Note first that $\overline{\varphi}_R(b+1,b) = b+1-p \geq 0 \geq \overline{\varphi}_C(b+1,b)$. The most complicated part is to show that $\overline{\varphi}_R(b+1,b) \geq \overline{\varphi}_M(b+1,b)$. To do this, we write

$$\begin{aligned}\overline{\varphi}_M(b+1,b) &= p\overline{\varphi}(b+2,b) + (1-p)\overline{\varphi}(b+1,b+1) - p(1-p) \\ &= p((b+2)\lambda - b\mu - c) + (1-p)\varphi(b+1,b+1) - p(1-p).\end{aligned}$$

From Corollary 1 for the immediate-release case, we get the bound $\varphi(b+1,b+1) \leq (b+1)p/(1-p)$. By substituting this we bound $\overline{\varphi}_M(b+1,b)$ from above by $2bp + p(2-4p+p^2)/(1-2p)$. We want this to be at most equal to $\overline{\varphi}_R(b+1,b) = b+1-p$ which gives the following inequality,

$$b(1-2p)^2 + 1 - 5p + 6p^2 - p^3 \geq 0.$$

Since $b \geq 0$, it suffices to have $1 - 5p + 6p^2 - p^3 \geq 0$ which holds for $p \leq 0.308$ (root of the polynomial $p^3 - 6p^2 + 5p - 1$). □

We now present the main result of this section based on the previous lemma.

Proof of Theorem 3. The above lemma implies that for every state (a, b),

$$\hat{g}_k(a,b) \leq k \cdot p + \overline{\varphi}(a,b). \tag{8}$$

Intuitively, $\overline{\varphi}(a,b)$ can only overestimate the optimal potential when $a \geq b+1$, even when $\hat{g}^* > p$.

We prove this using induction on k, b, and a. This is possible because Recurrence (5) of $\hat{g}_k$ depends on $\hat{g}_k(0,s)$ for $s < b$ (for $b > 0$), on $g_k(a+1,b)$, and/or $\hat{g}_{k-1}$.

For the outer induction on k, the base case $k = 0$ is trivial since $\hat{g}_0(a,b) = 0$ and $\overline{\varphi}(a,b) \geq 0$.

For a fixed k, we use double (strong) induction on b and (backwards induction on) a. So, for fixed k and b we assume that the statement holds for all $k' < k$ and for all states (a, b') such that $b' < b$. Note that for the base b=0, the induction step does not use the inductive hypothesis on b since a rational miner never capitulates from a state $(a, 0)$.

Since $\hat{g}_k(a,b)$ cannot be bigger than $k+b$, there is a value of a large enough such that $a\lambda - b\mu - c \geq k+b$; call this value $a_m(k,b)$. Statement (8) holds for every $a \geq a_m(k,b)$ because

$$\hat{g}_k(a,b) \leq k+b \leq a\lambda - b\mu - c = \overline{\varphi}(a,b) \leq k \cdot p + \overline{\varphi}(a,b).$$

We can then use any value $\hat{a} \geq a_m(k,b)$ as base case of the backwards induction on a. More formally,

$$\begin{aligned}\hat{g}_k(a,b) =& \max\Big\{\max_{s=0,\ldots,b-1} \hat{g}_k(0,s), p\,\hat{g}_k(a+1,b) + (1-p)\,\hat{g}_{k-1}(a,b+1), \\ & \qquad \hat{g}_{k-1}(a-b-1,0) + b + 1\Big\} \\ \leq& kp + \max\Big\{\max_{s=0,\ldots,b-1} \overline{\varphi}(0,s), p\,\overline{\varphi}(a+1,b) + (1-p)\,\overline{\varphi}(a,b+1) - p(1-p), \\ & \qquad \overline{\varphi}(a-b-1,0) + b + 1 - p\Big\} \\ =& kp + \overline{\varphi}(a,b),\end{aligned}$$

where the inequality holds by our induction hypothesis, and the last equality by Lemma 7.

Statement (8) shows that the optimal gain per step cannot exceed p, which shows that FRONTIER is best response for Miner 1. The proof of the theorem is complete. □

References

[Aspnes et al., 2005] Aspnes, J., Jackson, C., and Krishnamurthy, A. (2005). Exposing computationally-challenged Byzantine impostors. Technical Report YALEU/DCS/TR-1332, Yale University Department of Computer Science.

[Babaioff et al., 2012] Babaioff, M., Dobzinski, S., Oren, S., and Zohar, A. (2012). On bitcoin and red balloons. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 56–73, New York, NY, USA. ACM.

[Back, 1997] Back, A. (1997). Hashcash.

[Bonneau et al., 2015] Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., and Felten, E. W. (2015). Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *36th IEEE Symposium on Security & Privacy*.

[Courtois and Bahack, 2014] Courtois, N. T. and Bahack, L. (2014). On subversive miner strategies and block withholding attack in bitcoin digital currency. In *arXiv 1402.1718*.

[Dai, 1998] Dai, W. (1998). b-money.

[Dwork and Naor, 1993] Dwork, C. and Naor, M. (1993). Pricing via processing or combatting junk mail. In *Advances in Cryptology — CRYPTO' 92: 12th Annual International Cryptology Conference Santa Barbara, California, USA August 16–20, 1992 Proceedings*, pages 139–147, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Eyal, 2014] Eyal, I. (2014). The miner's dilemma. In *arXiv 1411.7099*.

[Eyal et al., 2014] Eyal, I., Gencer, A. E., Sirer, E. G., and van Renesse, R. (2014). Bitcoin-ng: A scalable blockchain protocol. In *arXiv 1510.02037*.

[Eyal and Sirer, 2014] Eyal, I. and Sirer, E. G. (2014). Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, pages 436–454, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Fergal Reid, 2012] Fergal Reid, M. H. (2012). An analysis of anonymity in the bitcoin system. In *Security and Privacy in Social Networks*, pages 197–223.

[Finney, 2011] Finney, H. (2011). Best practice for fast transaction acceptance - how high is the risk?

[Garay et al., 2015] Garay, J. A., Kiayias, A., and Leonardos, N. (2015). The bitcoin backbone protocol: Analysis and applications. In *Proceedings of Eurocrypt 2015*.

[Haber and Stornetta, 1991] Haber, S. and Stornetta, W. S. (1991). How to time-stamp a digital document. *Journal of Cryptology*, 3(2):99–111.

[Heilman et al., 2015] Heilman, E., Kendler, A., Zohar, A., and Goldberg, S. (2015). Eclipse attacks on bitcoin's peer-to-peer network. In *Proceedings of the 24th USENIX Conference on Security Symposium*, SEC'15, pages 129–144, Berkeley, CA, USA. USENIX Association.

[Juels and Brainard, 1999] Juels, A. and Brainard, J. (1999). Client puzzles: A cryptographic countermeasure against connection depletion attacks. In *Proceedings of NDSS '99 (Networks and Distributed Security Systems)*, pages 151–165.

[Karame et al., 2012] Karame, G. O., Androulaki, E., and Capkun, S. (2012). Double-spending fast payments in bitcoin. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 906–917, New York, NY, USA. ACM.

[Kroll et al., 2013] Kroll, J. A., Davey, I. C., and Felten, E. W. (2013). The economics of bitcoin mining, or bitcoin in the presence of adversaries.

[Lewenberg et al., 2015] Lewenberg, Y., Bachrach, Y., Sompolinsky, Y., Zohar, A., and Rosenschein, J. S. (2015). Bitcoin mining pools: A cooperative game theoretic analysis. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 919–927, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Meiklejohn et al., 2013] Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., and Savage, S. (2013). A fistful of bitcoins: Characterizing payments among men with no names. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 127–140, New York, NY, USA. ACM.

[Nakamoto, 2008] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

[Okun and Barak, 2007] Okun, M. and Barak, A. (2007). Efficient algorithms for anonymous byzantine agreement. *Theory of Computing Systems*, 42(2):222–238.

[Pease et al., 1980] Pease, M., Shostak, R., and Lamport, L. (1980). Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234.

[Rivest et al., 1996] Rivest, R. L., Shamir, A., and Wagner, D. A. (1996). Time-lock puzzles and timed-release crypto. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA.

[Ron and Shamir, 2013] Ron, D. and Shamir, A. (2013). Quantitative analysis of the full bitcoin transaction graph. In *Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers*, pages 6–24, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Rosenfeld, 2011] Rosenfeld, M. (2011). Analysis of bitcoin pooled mining reward systems. In *arXiv 1112.4980*.

[Rosenfeld, 2014] Rosenfeld, M. (2014). Analysis of hashrate-based double spending. In *arXiv 1402.2009*.

[Sapirstein et al., 2016] Sapirstein, A., Sompolinsky, Y., and Zohar, A. (2016). Optimal selfish mining strategies in bitcoin. In *Financial Cryptography 2016. To appear*.

[Sompolinsky and Zohar, 2015] Sompolinsky, Y. and Zohar, A. (2015). Secure high-rate transaction processing in bitcoin. In *Financial Cryptography and Data Security: 19th International Conference, FC 2015*.

[Tschorsch and Scheuermann, 2015] Tschorsch, F. and Scheuermann, B. (2015). Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. Technical Report 2015/464, Cryptology ePrint Archive.